

TIZEN™ DEVELOPER CONFERENCE MAY 7-9, 2012



Enabling Cordova (aka PhoneGap) on Tizen

René Pourtier / Luc Yriarte

What is Cordova (aka PhoneGap)?

- An **open-source standards-based development framework** for building cross-platform mobile applications **using HTML, CSS and JavaScript**

- An attempt to **unify third-party application development** across different mobile platforms and to **supplement existing standards with new JavaScript APIs for mobile development**

What is Cordova (aka PhoneGap)?

- A means to **author native applications with web technologies**
- A way to **deploy web apps on multiple platforms**. This can be done using the PhoneGap build service that creates app-store ready apps : « *write once, compile in the cloud, run anywhere* »

Official web site : www.phonegap.com

Cordova, a bit of history

- Initially called **PhoneGap**, i.e. bridging the Web API gap for Phone. It was later renamed **Cordova**
- Founded by the **Nitobi Software** company, then acquired by **Adobe** in Oct. 2011. Open-source licensing model is **Apache 2.0**
- Applied to **contribute to the Apache Software Foundation** and currently in the Incubation stage : see <http://incubator.apache.org/cordova/>

Cordova – Supported features

	iOS	Android	RIM OS	WebOS	WP7	Symbian	Bada
Accelerometer	✓	✓	✓	✓	✓	✓	✓
Camera	✓	✓	✓	✓	✓	✓	✓
Compass	✓	✓	✓				✓
Contacts	✓	✓	✓		✓	✓	✓
Files	✓	✓	✓		✓		✓
Geolocation	✓	✓	✓	✓	✓	✓	✓
Media	✓	✓			✓		
Network	✓	✓	✓	✓	✓	✓	✓
Notification-alert	✓	✓	✓	✓	✓	✓	✓
Notification-sound	✓	✓	✓	✓	✓	✓	✓
Notification-vibrate	✓	✓	✓	✓	✓	✓	✓
Storage	✓	✓	✓	✓	✓	✓	

Source : <http://phonegap.com/about/features>

Why port Cordova to Tizen?

- **Most popular cross-platform way of developing and packaging hybrid apps**
- **Establish Tizen as a global open-source web platform**
- **Allow Tizen devices to run the Cordova apps portfolio and to leverage the Cordova ecosystem**
- **Benefit from the PhoneGap build system's ability to deploy web apps on other platforms supported by Cordova**

Main reference implementations

- **Native platform code with HTML5/JS/CSS code inside**
 - fullscreen WebView instantiated from native code with mechanisms to bridge between native code <-> javascript code
 - bridge mechanisms depending on platform WebView implementations features
 - examples : iOS (ObjectiveC and WebView Control + gap:// protocol), Android (Java and WebView Control + gap:// protocol), BlackBerry (Java and WebView Control + BlackBerry WebWorks JavaScript Extension)
- **JavaScript shim layer**

JavaScript shim layer

We opted for a shim layer = set of JavaScript libraries mapping/wrapping Cordova Web API on top of Tizen Web APIs

- **Pros**

- relies on the Tizen public SDK
- easier to implement thin adaptation layer on top of Tizen
- convenient to package, build and test (Tizen .wgt format)
- proven by WebOS and Symbian

- **Cons**

- limited to the capabilities exposed by Tizen Web APIs
- no extensibility to new platform services

Cordova and Tizen Web APIs mapping

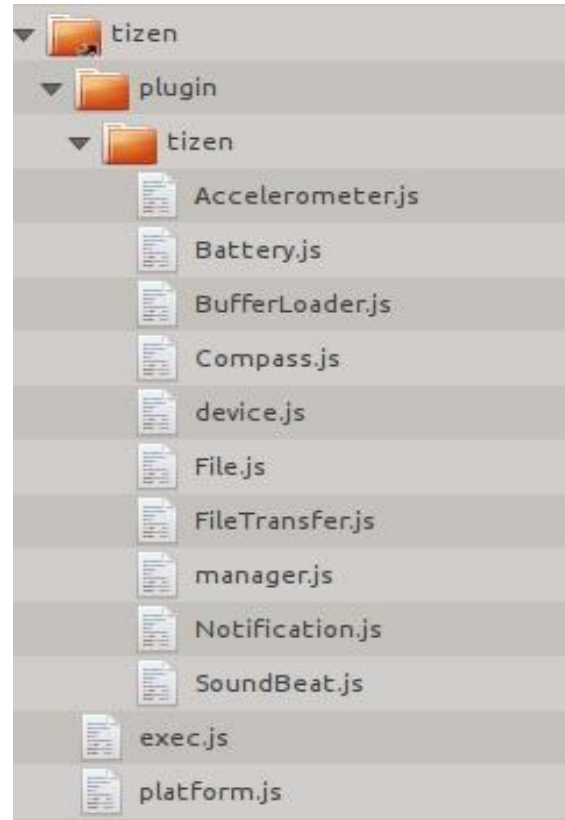
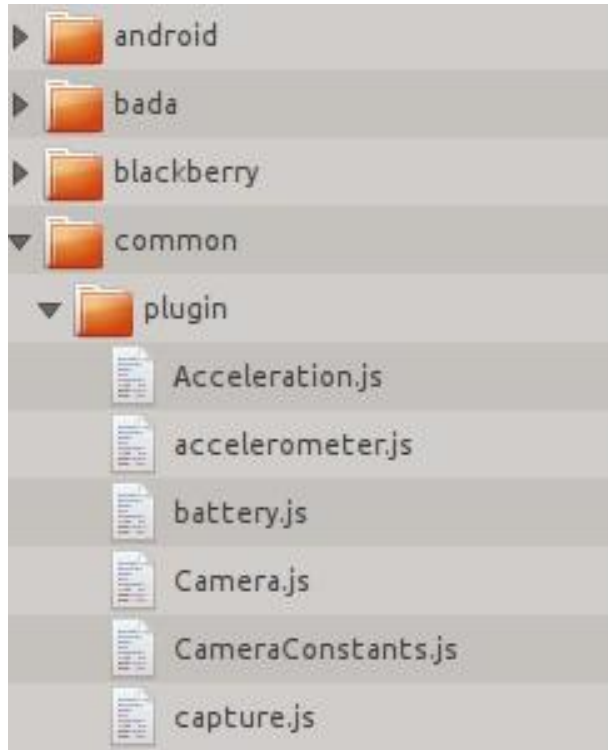
Cordova	Tizen
Accelerometer	W3C Device Orientation API
Compass	W3C Device Orientation API
Geolocation	W3C Geolocation API
Device	Tizen System Information API
Connection	W3C Network Information API
Storage	W3C WebStorage/WebSQL API
Files	W3C File/Directories/Writer API
Notifications	HTML5/W3C Vibration API
Events	Tizen System Information API (Battery...)
Contacts	Tizen Contacts Web API
Camera	Tizen Application Web API or W3C HTML5 Media Capture
Capture	Tizen Application Web API or W3C HTML5 Media Capture
Media	Tizen Application Web API or W3C HTML5 Audio

Cordova sample application

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <script src="cordova.js"></script>
  </head>

  <body>
    <script>
      document.addEventListener('deviceready', function () {
        console.log("Cordova is ready for platform " + window.device.platform + "
v" + window.device.version + "!");
        // Cordova is ready, do all your Cordova stuff here
      }, false);
    </script>
  </body>
</html>
```

Common code VS host specific code



Generic Cordova JS code (common)

```
// ....full code not included....
```

```
Battery.prototype.onSubscribe = function() {
```

```
  var me = battery;
```

```
  // If we just registered the first handler, make sure native listener is started.
```

```
  if (handlers() === 1)
```

```
    exec(me._status, me._error, "Battery", "start", []);
```

```
};
```

```
Battery.prototype.onUnsubscribe = function() {
```

```
  var me = battery;
```

```
  // If we just unregistered the last handler, make sure native listener is stopped.
```

```
  if (handlers() === 0)
```

```
    exec(null, null, "Battery", "stop", []);
```

```
};
```

Battery shim layer (Tizen-specific)

```
var id = null;
module.exports = {
  start: function(successCB, failCB) {
    var tizenSuccessCB = function(power) {
      if (successCB)
        successCB({level: Math.round(power.level * 100), isPlugged: power.isCharging});
    };
    if (id === null)
      id = tizen.systeminfo.addPropertyChangeListener("Power",
tizenSuccessCB);
tizen.systeminfo.getPropertyValue("Power", tizenSuccessCB, failCB);
  },
  stop: function(successCB, failCB) {
    tizen.systeminfo.removePropertyChangeListener(id);
    id = null;  }  };
}
```

JS plug-in manager (Tizen-specific)

```
var cordova = require('cordova');
module.exports = {
  exec: function (successCB, failCB, clazz, action, args) {
    var plugin = require('cordova/plugin/tizen/' + clazz);
    if (plugin && typeof plugin[action] === 'function') {
      var result = plugin[action](successCB, failCB, args);
      return result || {status: cordova.callbackStatus.NO_RESULT};
    }
    return {"status" :
cordova.callbackStatus.CLASS_NOT_FOUND_EXCEPTION, "message" :
"Function " + clazz + "::" + action + " cannot be found"};
  }
};
```

Implementing as native hybrid apps

- **Native Tizen EFL application** instantiating an **Elementary web view widget** that renders/executes the Cordova web app embedded in its resources (HTML/JavaScript/CSS)
- **Bridge mechanism** and **protocol** allowing interactions between JavaScript and native code in both directions.

- **JS ► Native:** Document title changed hook native callback
- **Native ► JS:** `ewk_frame_script_execute(webkit, javascript_code)`
- **Gap protocol:** `gap://Service/Action/Request_id/[JSON arguments]`

- **Cordova API services** implementation based on **native Tizen SDK** (instead of Tizen Web API SDK)

JS -> Native: exec.js

```
module.exports = function(success, fail, service, action, args) {  
  try {  
    // Generate a command unique ID (transaction ID)  
    var cmdId = service + cordova.callbackId++;  
    // Register the command callbacks (to be called by native side)  
    if (success || fail)  
      cordova.callbacks[cmdId] = {success:success, fail:fail};  
    // Build gap URI (gap://Accelerometer/getAcceleration/Accelerometer0/[])  
    var command = service + "/" + action + "/" + cmdId + "/" + JSON.stringify(args);  
    // Change the document title to trigger our hook callback on native side  
    document.title = "gap://" + command;  
    return null;  
  } catch (e) { utils.alert("Error: " + e); }  
};
```


Native -> JS

Once processed by native side, a command request reply is sent back to the JS side by asking its registered JS callbacks execution.

```
cordova.callbackSuccess(<callback_id>,  
    {status:<status>,message:<result_message>,keepCallback:<true_or_false>});
```

```
cordova.callbacksError(<callback_id>,  
    {code:<error_code>,message:<error_message>});
```

Native side:

```
ewk_frame_script_execute(main_frame,  
"cordova.callbackSuccess('Accelerometer0',{status:'1',message:{x:1.23,y:4.56,z  
:7.89}, keepCallback:'false'});");
```

Current status

- **Code has been public** since 03/13, as announced on the Cordova mailing list
 - <https://github.com/otcshare/cordova-tizen>
- **Cordova Hybrid Application feasibility completed**
 - prototype done
 - pending questions about security policy

Current status

- **APIs fully ported (shim layer)**
 - Accelerometer, Compass, Geolocation, Device, Storage, Files, Notifications, Power events

- **APIs in progress (shim layer)**
 - Events, Camera, Contacts, Capture, Media

Next steps

- **Complete the missing APIs**
 - switch/upgrade to SDK 1.0 Larkspur
 - try to hit June target

- **Get Cordova Tizen project hosted by the Apache Software Foundation**
 - will require to sign ICLA/CLA
 - will need approval from Cordova « mentors »

Next steps

- **Complete the support of Cordova Hybrid applications**
 - clarify strategy about native Tizen development kit/native apps
 - complete prototype to replace shim layer (TBC)
- **Add Cordova Tizen build capabilities** to PhoneGap build services
 - once our port to Tizen is completed and reliable
 - meet with Cordova people to put it in place

Time for a demo

Thanks for your attention!

Q & A

Contacts for Dev team

Email us at :

paul.plaquette@intel.com

Intel Software Engineer

regis.merlino@intel.com

Intel Software Engineer

christophe.guiraud@intel.com

Intel Software Engineer

rene.pourtier@intel.com

Intel Engineering Manager

TIZEN™ DEVELOPER
CONFERENCE
MAY 7-9, 2012